Mississippi State University's Entry for the 2008 AUVSI Student UAS Competition

Christopher Edwards
Team Lead

Marty Brennan
Airframe Lead

Brittany Penland
Systems Lead

XIPITER
MISSISSIPPI STATE UNIVERSITY

ABSTRACT

The 2008 Student UAS Competition marks Mississippi State University's fifth year of participation. Team Xipiter has taken an evolutionary approach to accomplish mission objectives involved with gathering and delivering intelligence, surveillance, and reconnaissance (ISR). The X-2C Xawk UAS couples a robust airframe, designed and built by students, with a combination of commercial off-the-shelf (COTS) hardware and student-designed software components into a dynamic system capable of gathering surveillance during fully autonomous flight. The airframe is fabricated using preimpregnated carbon/fiberglass hybrid material and carries a payload of up to 25 lbs. Onboard components of the system include an autopilot, a digital pan/tilt/zoom camera, and a broadband ethernet bridge. On the ground, software controls the autopilot and surveillance system. To ensure safety, a flight control redundancy system has been installed to allow two receiver and power inputs. With one receiver connected with the autopilot and the other connected directly to an R/C receiver, the pilot is able to completely remove the autopilot system from the control loop.

1. TEAM DYNAMIC

Team Xipiter is a multidisciplinary design team consisting of seventeen members. Students range in classification from freshmen to graduate and encompass five majors. By utilizing members' education from Aerospace Engineering, Electrical Engineering, Computer Engineering, Computer Science Engineering, Biological Engineering, and Elementary Education, the team is able to capitalize on each person's experience and knowledge to better develop a complete system. This diversity strengthens the Team Xipiter and sets it apart in competition.

2. CONSTRAINTS AND REQUIREMENTS

The design considerations observed by Team Xipiter reflect the requirements outlined in the Student UAS Competition rules. The constraints governing the design, production, and integration of the X-2C system are detailed in this section. The team is allowed up to 40 minutes to complete the mission from start to finish. The objective is to conclude the mission within 20 minutes and deliver target identification in real time.

2.1 Air Vehicle

In addition to being capable of heavier than air flight, a number of technical constraints are placed on the air vehicle. Aircraft Modelers Association (AMA) requirements must be met as well as complying with regulations in takeoff weight and flight in certain environmental conditions. The air vehicle must have a gross takeoff weight of less than 55 lbs with full systems included. The airframe must also be capable of takeoff and landing in crosswinds of 8 kts, continuous flight in winds of 15 kts, and operation in temperatures up to 110 °F at 1000 ft MSL.

2.2 Systems

The system design is directed by key performance parameters which include autonomy, in-flight re-tasking, imagery, target location, and mission time. The system must be autonomous during waypoint navigation with an additional objective of autonomous takeoff and landing. The system must also be capable of modifying waypoints and search areas during flight. Target imagery must be displayed to the judges with identification of the five target parameters: shape, background color, orientation, alphanumeric, and alphanumeric color. Target location must be determined within 250 ft with the objective of within 50 ft.

2.3 Safety

Safety is an important factor in the mission requirements. In order to verify structural integrity of the airframe, all components must be secured and fastened properly, correct wiring must be installed for payload requirements, and communication range must be verified. The system must be equipped with fail-safe modes. The pilot must be able to take control of the vehicle at any point during flight, and the system must be capable of activating a flight termination sequence upon command or following the loss of communication link.

3. DESIGN

3.1 Air Vehicle

The X-2C airframe is the product of an evolutionary approach. Improvements have included increased size, addition of flaps, pusher engine configuration, increased engine performance, specially designed landing gear system, and removable nose cone. The design is based on the X-2A airframe, but has undergone modifications with emphasis on cruise performance and integration of systems. The larger wing span and addition of flaps allow the plane to fly more slowly and with greater stability. The longer fuselage results in more internal space for easier access to the payload. The airframe was fabricated using preimpregnated carbon/fiberglass hybrid material for the fuselage, wings, and empennage, and COTS carbon tubes for the spars and booms. The 100 cc 2-cylinder engine, more powerful than previous years' single cylinder engines, provides additional maneuverability during low-speed flight, increased performance in cruise conditions, and less induced vibration. The engine is mounted in a pusher configuration to eliminate residue obstruction of surveillance and mitigate vibration to the onboard systems. The landing gear system has been custom designed to better withstand taxi and landing loads without diminishing maneuverability. A removable nose cone provides access to additional working space and the nose gear.

3.1.1 Evolutionary Approach

Team Xipiter has taken an evolutionary approach to developing a reliable UAS for use in competition as well as for real world applications. X-2C, the air vehicle for this year's competition, has been preceded by Senior Telemaster, X-1, X-2A, and X-2B.

The airframe used for the 2004 AUVSI UAV Competition was an off-the-shelf model, which made construction easy. The Senior Telemaster (Figure 1) had a tail dragger configuration and a split horizontal stabilizer. The engine was in a tractor configuration and required glow fuel. The team concluded that the Senior Telemaster, despite its easy construction, had insufficient internal room for the payload.

To provide more payload volume, Team Xipiter designed X-1 (Figure 2) for the 2005 competition. This airframe was fabricated using fiberglass with a wet lay-up fabrication technique. X-1 had a tricycle landing gear configuration and the gasoline engine was placed in a tractor configuration. The X-1 airframe had a much larger internal volume than the Senior Telemaster. However, new challenges surfaced with X-1's airframe. Mainly, the engine placement restricted access to the payload, and exhaust residue interfered with the camera surveillance system.

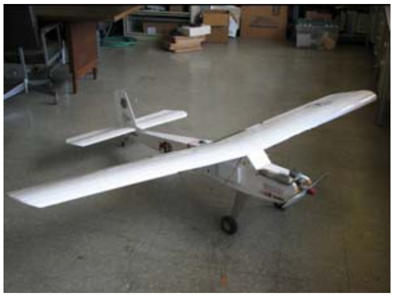


Figure 1. Senior Telemaster

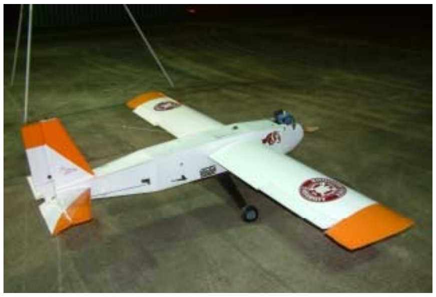


Figure 2. X-1 Air Vehicle

The X-2A airframe (Figure 3) was used in the 2006 competition and addressed problems associated with the X-1 air vehicle. X-2A was constructed of preimpregnated carbon fiber using student-built carbon fiber molds, simplifying the manufacturing process and helping to decrease the gross weight of the air vehicle. A pusher engine configuration was chosen to eliminate surveillance interference and to allow greater access to the payload area. While problems associated with the X-1 air vehicle were solved, X-2A's surveillance capability suffered due to a high cruise speed.

Built for the 2007 competition, the X-2B airframe (Figure 4) evolved from the design of X-2A. Changes included an increased wingspan and lengthened booms, resulting in a decreased minimum flight speed and greater stability. Even greater internal volume was achieved by lengthening the fuselage. A brake system was also installed to minimize the aircraft's landing distance and to ensure safer ground taxi operation.

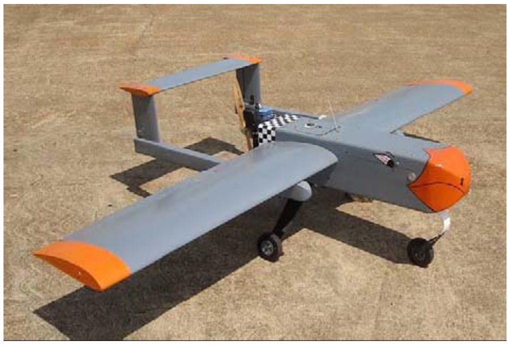


Figure 3. X-2A Airframe

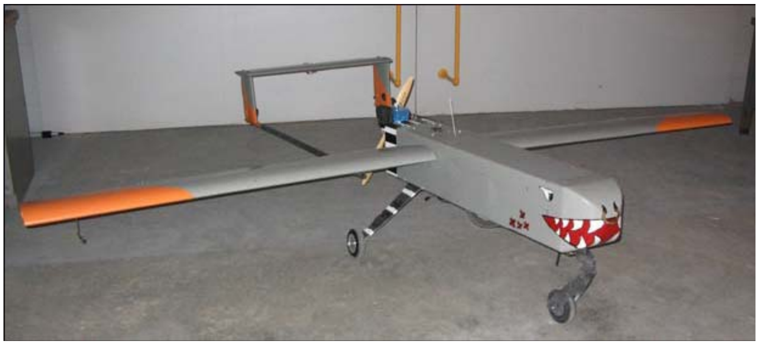


Figure 4. X-2B Air Vehicle

3.1.2 Air Vehicle Design

The X-2C air vehicle (Figure 5) has an improved design from X-2A and X-2B. Although the basic design was retained, modifications in fabrication materials, engine type, landing gear, and nose cone have enhanced the performance of the X-2C UAS.

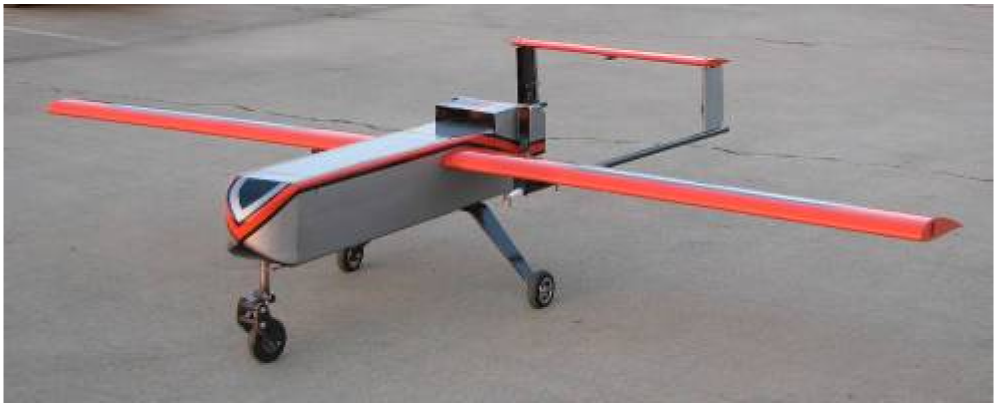


Figure 5. X-2C Air Vehicle

The flight characteristics of X-2B were effective in cruise and slow flight, thus the wing design remained mostly unchanged for the X-2C air vehicle. The wings employ a SD7062 airfoil. In order to enhance slow-flight maneuverability, the chord of the control surfaces on the wings was increased. Figure 6 provides a planform view of a wing. The fuselage of X-2C preserved the same dimensions as the X-2B air vehicle. Wing and fuselage dimensions are shown in Table 1.

Table 1. X-2C Wing and Fuselage Dimensions

	Parameter	*Value*
Wings	Span	128 in
	Chord	16 in
	Area	14.22 ft^2
	Aspect Ratio	8
	Control Surfaces	4 in x 21 in
Fuselage	Length	45 in
	Width	9 in
	Height	9 in
	Internal Volume	2.11 ft^3

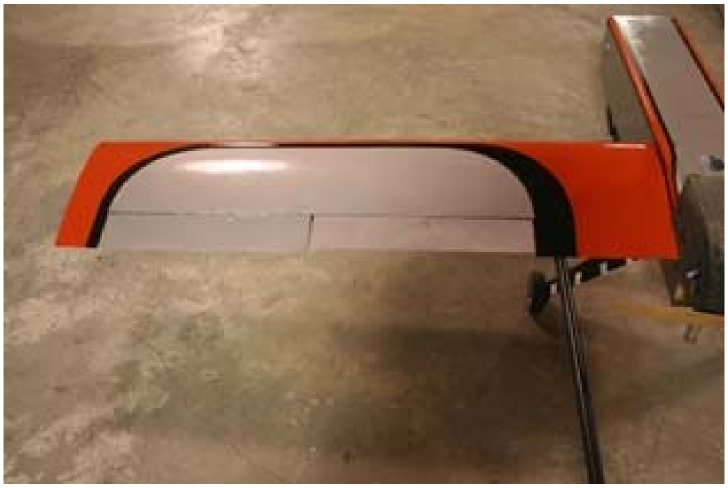


Figure 6. Planform View of a Wing

A structural addition to the X-2C airframe is the use of right-angle brackets to connect load-bearing members. In the wing, the ribs were joined to the wing skins using a single piece called a "pi-clip" (Figure 7). The pi-clips were used on both the upper and lower wing skins to distribute loads to and from the ribs more effectively than a quarter-inch edge bond used for previous airframes. In the fuselage, pieces of carbon-fiber right-angle stock were used to join the bulkheads to the flat sides of the fuselage. The right-angle stock was also used as a longeron on the left and right sides of the fuselage. The longerons greatly increased the rigidity of the fuselage and provided a convenient location on which to mount payload components and run wires.

A new nose gear design features a trailing link shock-absorbing system that damps minor shocks during taxi testing and absorbs large landing loads without any loss in ground maneuverability. This gear requires a removable nose cone (Figure 8) to provide access to the area forward of the front bulkhead. The removable nose cone also allows batteries to be placed the greatest distance forward, thus helping to increase the static margin. The removable nose cone is similar to nose cones of previous X-2 air vehicles but attaches to the fuselage with two fasteners on each side.

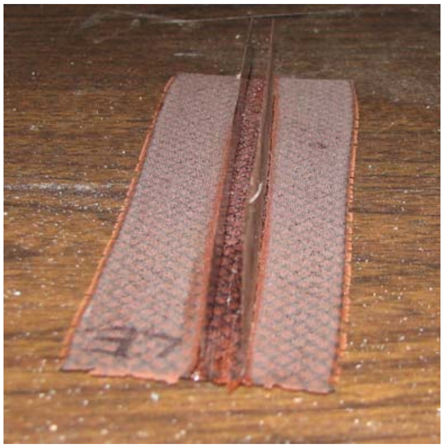


Figure 7. Pi-clips Used in Wings

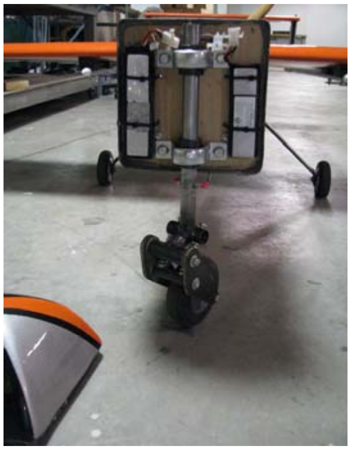


Figure 8. Removable Nose cone

The empennage of the X-2C air vehicle is mounted to the wings by twin booms and consists of twin vertical stabilizers joined by a horizontal stabilizer in a pi configuration. The J5012 airfoil is used for both the vertical and horizontal stabilizers. The dimensions for the empennage are presented in Table 2.

Table 2. X-2C Empennage Dimensions

Parameter	*Vertical*	*Horizontal*
Height/Span	7 in	33 in
Chord	9.25 in	9 in
Area	129.5 in^2	297 in^2
Aspect Ratio	1.51	3.67

3.1.3 Air Vehicle Fabrication

Using previously manufactured molds, the upper and lower wing skins are made of preimpregnated carbon/fiberglass hybrid, referred to henceforth as hybrid, material with a Divinycell foam core. Each of the wing ribs is made of 0.25 in birch sandwiched between layers of the hybrid material. The wing spar is a COTS 1.5 in diameter carbon tube placed at the quarter chord of the wing. The spar has been reinforced at the root by inserting a solid nylon rod. The anti-torque rear spar is a COTS 0.25 in solid carbon rod placed 9.5 in from the leading edge of the wing. Leading edges for the wings are made of the hybrid material and are formed using previously manufactured molds. The boom attachments are reinforced with 2 ribs on either side of the attachment locations. This construction supplies additional structural rigidity at the high stress locations of the boom attachments.

The fuselage of X-2C is made of a sandwich construction of hybrid and a Divinycell foam core. The three hatches and the nosecone are composed solely of hybrid. The three bulkheads that separate the main payload compartment, the autopilot compartment, and the rear compartment are composed of a sandwich construction of 0.25 in oak and hybrid material. Two wing-fuselage attachments one placed on the second bulkhead and the other on the third bulkhead, secure the wings in the fuselage in a male/female carry-through fashion using locking pins. These wing-fuselage attachments were fabricated using preimpregnated carbon.

The pi-clips joining the wing ribs to skins were fabricated individually from the hybrid material. Rib templates for both the upper and lower surfaces were cut from wood and covered with Teflon release tape. The hybrid material was placed into the wing-skin molds and formed around the rib templates, which remained in place during the oven cure time. The pi-clips were removed and dry-fitted to the wing skins and ribs before bonding. The fit was checked with clay balls (Figure 9) to ensure all surfaces were in contact within a close tolerance. The carbon-fiber right-angle stock was fabricated using aluminum angle as a template. A long piece of angle stock was cut to the desired lengths of the various components in the fuselage.

The empennage (Figure 10) is attached to the aircraft by two COTS 1.0 in diameter carbon tubes. The skins of the stabilizers are made using the hybrid with a balsa core formed in previously manufactured molds. Each vertical stabilizer is secured to a boom with bolts, and the horizontal stabilizer is fastened to the vertical stabilizers in a pi-configuration using nylon bolts.

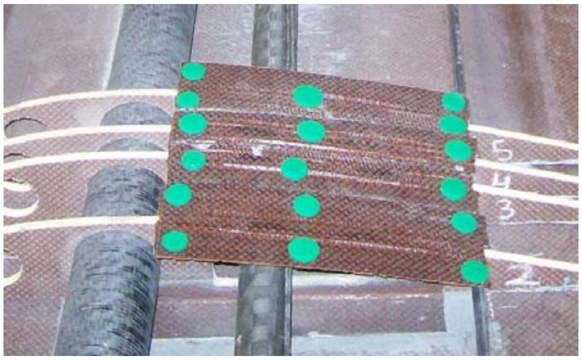


Figure 9. Clay Ball Test

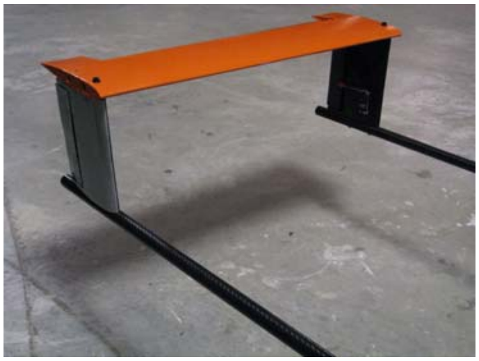


Figure 10. Empennage of X-2C

The landing gear of the X-2C air vehicle has a traditional tricycle configuration. The nose gear (Figure 11) was designed by Team Xipiter as a solution to structural and control issues found in past designs. The nose gear is composed of two carbon-fiber and birch wood side plates linking the wheel to a steel shaft and a spring-damping system. The main gear (Figure 12) was fabricated from uni- and bi-directional preimpregnated carbon fiber using an aluminum spring leaf mold.

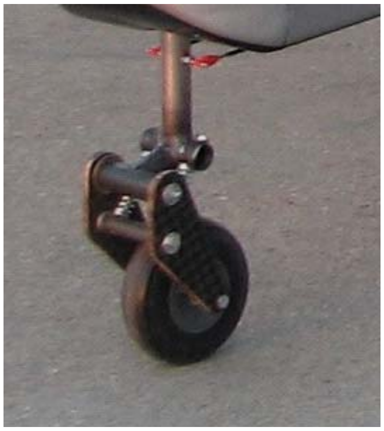


Figure 11. Nose Gear of X-2C

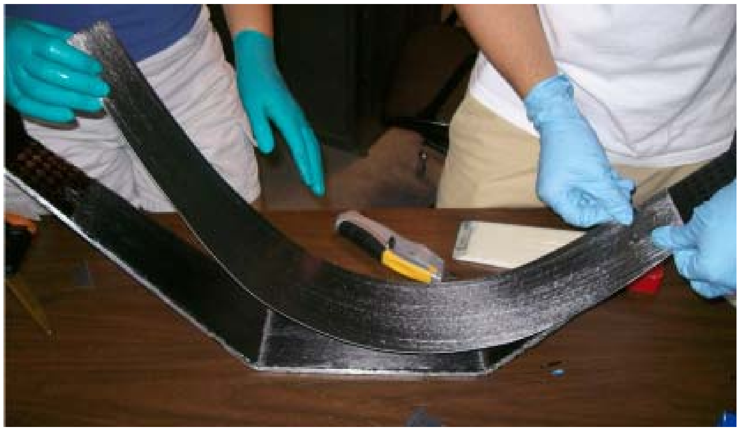


Figure 12. Main Landing Gear of X-2C

Reliability and performance are important factors in selecting an engine. An unintended shutdown of the engine during flight could lead to a rough landing, potentially damaging both the aircraft and the equipment on board. Designed specifically for model aircraft, Kroma engines deliver the high performance required for the X-2C air vehicle. The Kroma 100i 2-cylinder engine was chosen for its high power-to-weight ratio. More power was required to improve performance and control, especially during takeoff and landing. Table 3 displays specifications of the Kroma engine.

Table 3. X-2C Engine Specifications

Parameter	*Value*
Displacement	100 cc
RPM range	1,100-9,000 rpm
Output	10 hp @ 9,000 rpm
Weight (w/o mufflers)	5 lbs

The Kroma 100i (Figure 13) utilizes an electronic ignition (EI) system, powered by a 6 V battery, to provide ignition timing. Kroma engines require a mixture of gasoline and 2-stroke motor oil. The engine, placed in the rear of the fuselage in a pusher configuration, uses a Xoar tri-blade beechwood 26 in x 10 in pusher propeller. Because the engine is placed behind the fuselage, proper air flow over the cooling fins is blocked. To maintain an operational temperature during flight, a cowl was designed and fabricated to provide sufficient airflow for engine cooling.

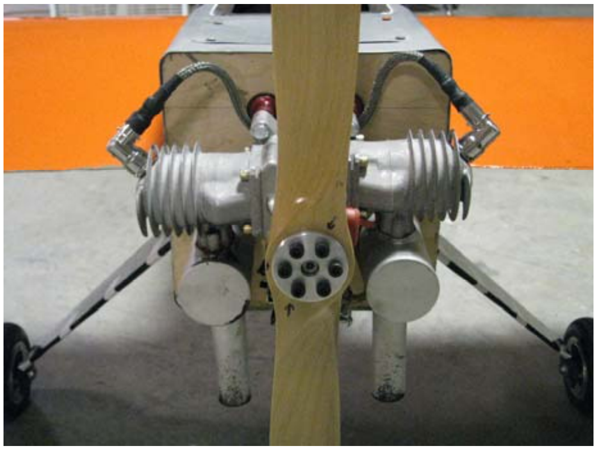


Figure 13. Kroma 100i Engine

3.2 Systems Hardware

Two critical systems in X-2C are the autopilot and the surveillance system. The autopilot is used as the primary flight control system. The surveillance system consists of the camera, battery, and ethernet bridge.

3.2.1 Autopilot

A thorough trade study was performed to determine the optimum autopilot for meeting the autonomy requirement for competition. The trade study evaluated the following systems: Micropilot 2028g, Cloud Cap Piccolo LT, UAV Navigation AP04R, Paparazzi, and Procerus

Kestrel. Important criteria considered for each system were size, weight, cost, autonomous capabilities, and power consumption. After analyzing the results, the Piccolo LT autopilot (Figure 14) from Cloud Cap Technologies was chosen for autonomous operation of the X-2C UAS.

The Piccolo LT system is capable of controlling up to 7 servo channels during all phases of flight. The autopilot also has several built-in safety features, including an R/C failsafe mode and a return-home command. The Piccolo LT communicates with the operator through the Piccolo Command Center, which provides information on aircraft location and orientation, in-depth telemetry, and systems data. The operator uses this interface to supply the autopilot with waypoint information.

Figure 14. Piccolo LT Autopilot

Cloud Cap provides the Athena Vortex Lattice (AVL), a program that uses a vortex-lattice method to determine stability derivatives in order to estimate the effectiveness of the aircraft. An aircraft model is created using AVL and is used in a provided simulator to generate gains for the autopilot control system. This simulator can also be used in conjunction with the Piccolo Command Center to perform software-in-the-loop (SIL) and hardware-in-the-loop (HIL) tests. These tests allow missions to be simulated before attempting test flights.

3.2.2 Surveillance System

For previous competitions, an analog camera was used to gather images of targets. Problems encountered using this system setup included susceptibility to noise and the need for extra hardware to control the camera and transmit data. These problems were addressed in a thorough trade study, in which the Sony D70, Toshiba IK-WB21A, and Bronzepoint PTZ 270 were considered. Criteria included cost, weight, resolution, zoom, and power consumption. The Toshiba digital network camera (Figure 15) was chosen for this year’s system. The camera exhibited pan/tilt/zoom capabilities, 22x optical zoom, and resolution of 1280 x 960 pixels.

Figure 15. Toshiba IK-WB21A Camera

For this setup, a wireless link capable of 10 Mbps is needed for communication between the camera and ground station. After some research, a Microhard wireless ethernet bridge was chosen. The use of this bridge provides several advantages of this system versus the system used during previous competitions. One advantage is that multiple devices can communicate via the same transceiver. Another is the use of digital communication instead of analog, greatly decreasing the effects of interference. Lastly, the practice of frequency hopping increases the reliability and security of the communication link. If significant interference is detected, the bridges change frequencies within a certain range until they find a frequency that has an acceptable amount of interference.

3.3 System Software

The system software consists of all of the software used for control, data reception, and data processing. This includes the Xipiter Camera Control Software (XCCS), auto target-recognition (XawkEyes), Xipiter Base Station (XBS), Joint Architecture for Unmanned Systems (JAUS), and Piccolo Command Center software.

3.3.1 Camera Control (XCCS)

This year's camera setup requires software that is capable of recording video, saving individual images, recording the plane state with each image, and providing control of the camera's settings and position. The software was designed to be user-friendly function efficiently, allow for easy modifications and extensions, and remain aesthetically-pleasing.

XCCS contains individual modules that communicate with the gimbal system, video devices, joysticks, and autopilots. For each of these categories, an interface was created to describe a general device. For example, a joystick module should have a method of retrieving the x, y, and z coordinates of the joystick. Using these interfaces, modules of code can be written specifically for one device. For example, there are two types of video devices that exist in the current library: a network camera video module and a DirectX video device. Each of these modules can be written to allow for the most efficient use via the interface methods.

To manage all of the smaller modules in a given category, a collection module exists. This collection module is compiled with a list of the different types of smaller modules in its category. Software code on a higher level communicates with this collection modules, not with the smaller modules below it. This allows for extra modules to be added easily. When a new device is introduced that needs to communicate with XCCS, a module is simply written using information from that device’s interface control document (ICD). Once this is done, the list in the collection module needs to be updated and recompiled.

3.3.2 Automatic Target Recognition (XawkEyes)

Several methods of detection were considered. One approach was to convert the color image to a gray scale equivalent and perform edge detection using an algorithm developed for black and white images. Examples of this edge detection method are Canny edge detection [1], Laplace indicators [2], and Sobel edges [3]. However, losses occur in the process of converting a color image to a gray scale image, causing different colors to become the same shade of gray. This loss of data causes inaccuracies in object recognition. Another option considered was color histograms, which classify color images [4, 5]. This system converts the target image into a histogram that represents the color of the target. The target histogram can then quickly be compared to histograms of portions of the search area. Matches are determined by similar histograms. This system matches images that have similar colors in similar proportions; however, histogram indexing contains no spatial information such as the shape of objects. This can cause inaccuracies and is sensitive to the sampling size of the searched image. Lastly, the method of edge color distribution was considered. Edge color distribution divides target recognition task into two steps [6]. Step one transforms the two-dimensional image into a three-dimensional color space. Step two detects objects in a single color space. Considering these possibilities, the edge color distribution method was deemed the most practical for this application.

The edge color distribution transformation takes the original two-dimensional image and converts it into a three-dimensional color space with the third dimension being color. In a 2-D image containing just black and white colors, the corresponding 3-D image has one color level for black and one color level for white with all white pixels represented on the white level. More colors simply add more levels to the 3-D space. Ideally, the objects to be recognized are mono-color, but various conditions, such as lighting and camera angle, can make the objects appear to be an aggregate of similar colors. A bucketing system aggregates several numeric values into one "color." The bucketing system can be used much like a color histogram to find objects that have similar colors in similar proportions. The advantage over color histograms is that spatial information is retained and can be used during object detection.

To detect an object, a shape recognizer based on the one developed by Schauf et. al [7] was employed. This recognizer was designed for a black and white environment but can be used on the mono-color planes in the 3-D space produced by XawkEyes. The recognizer "erodes" the shape in sixteen separate ways. The erosion process removes the outer edges of the shape and focuses on the center. This helps sharpen images with fuzzy edges or small occlusions by removing edge distortions. Erosion loses some details of the shapes so that similar shapes erode to the same basic geometry. Due to these losses, each candidate target shape is eroded all sixteen ways and compared to a library of shapes. The shape that obtains the most matches throughout the erosion process is the overall match and the shape compatibility percentage is reported. Additionally, shape distortion due to the viewing angle must be taken into account by having images from many different angles in the library.

3.3.3 Xipiter Base Station (XBS)

The XBS software (Figure 16) is a student-written application used for target identification and characterization. XBS is designed to be easy to use and flexible in time sensitive situations, such as competition. The XBS software user interface is implemented using National Instruments' LabVIEW, which provides a process for working with the pictures gathered by the X-2C UAS.

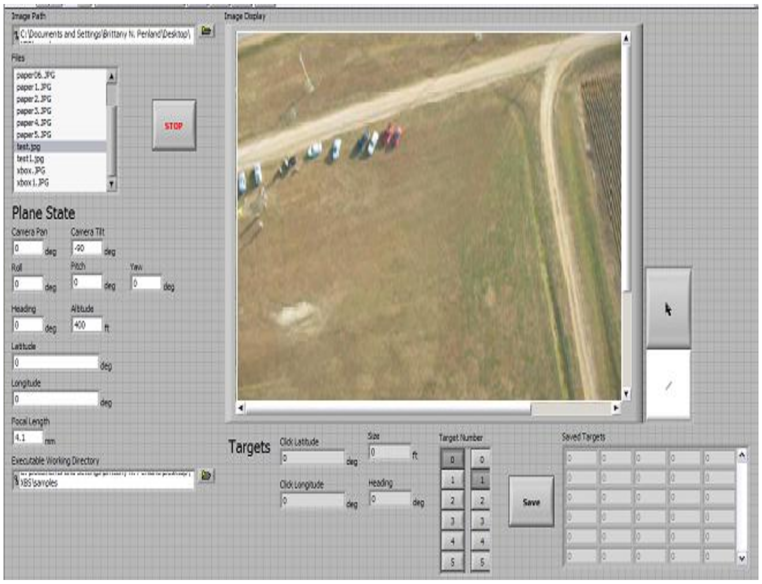


Image Path: C:\Documents and Settings\Brittany N. Penland\Desktop\

Files:
- paper06.JPG
- paper1.JPG
- paper2.JPG
- paper3.JPG
- paper4.JPG
- paper5.JPG
- test.jpg
- test1.jpg
- xbox.JPG
- xbox1.JPG

Plane State

Camera Pan: 0 deg
Camera Tilt: -90 deg
Roll: 0 deg
Pitch: 0 deg
Yaw: 0 deg
Heading: 0 deg
Altitude: 400 ft
Latitude: 0 deg
Longitude: 0 deg
Focal Length: 4.1 mm
Executable Working Directory: XBS\samples

Targets

Click Latitude: 0 deg
Size: 0 ft
Click Longitude: 0 deg
Heading: 0 deg
Target Number: 0 1 2 3 4 5 | 0 1 2 3 4 5

Saved Targets

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Figure 16. XBS Software

A picture can be selected from a list containing all of the images available in a shared folder between the XBS laptop and the video laptop. Targets can then be selected by the user via a cursor or line tool. Information about the target, such as GPS coordinates, length, and heading, can be derived using the plane state data obtained from the autopilot at the time the picture was taken. Finally, this information can be saved for later analysis by assigning the target a unique number.

The Xipiter Photogrammetry Calculator (XPC) software is the calculation core of the base station software and is written in Python with the associated NumPy add-on module. The mathematics behind XPC are based on photogrammetry, or the science of obtaining information about the real world from photography and other measurements [8]. XPC begins by establishing a series of six coordinate systems that fully describe the position and orientation of the aircraft and the camera. The coordinate systems are a local ground-parallel system with axes parallel to the standard directional axes, the aircraft wind axis system that takes into account angle of attack and sideslip; the aircraft body system; a camera base system for pan; a camera body system for tilt; and a final photograph system with the x-axis out the right side of the image and the y-axis out the top of the image for the photogrammetric calculations themselves. All systems are assumed to originate at the center of gravity of the aircraft. A series of five three-dimensional coordinate transformation matrices are developed by calculating a series of rotation angles from one coordinate system to the next. These matrices are then multiplied together to give a single resultant transformation matrix from the ground-parallel system directly to the photograph system. Since all photogrammetric calculations are in terms of the angles of tilt, swing, and azimuth, a second set of calculations are done to convert the resulting angles to the final tilt-swing-azimuth system.

After the flight is complete and a full set of targets have been identified and saved to the on-screen list, an "Average" button is available for post processing. When pressed, the Average button combines all data sets corresponding to the same target number and averages the data, giving one final result for each target. Assuming multiple observations are made in which the user can identify the target by number, any numberless data sets are discarded as unnecessary. Once processed, the resulting list can be saved to the computer as a CSV file. This works for both averaged and non-averaged data, so if the previous assumption is incorrect, the raw data can be saved for later processing.

3.3.4 Joint Architecture for Unmanned Systems (JAUS)

JAUS implementation was new this year for the competition. Programming languages, Java and C++ were compared, as to which would be best for this systems components. Java was chosen because of the built in data type, BitSet, to handle bit arrays. After the requirements for JAUS were analyzed and the programming language chosen, JAUS implementation could be researched further. Resources such as the official documentation and several open source projects, such as OpenJUAS, were investigated to see if they were usable and to help get a better idea of implementing the JAUS protocol. Because of the AUVSI requirements, it was determined that starting afresh would be more time and resource efficient. The software structure is shown below in Figure 17.

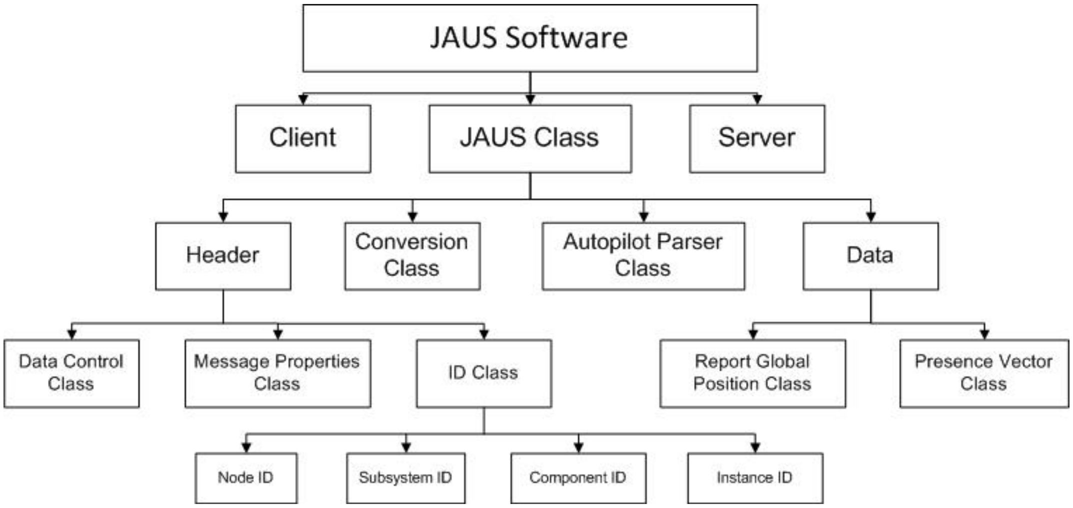


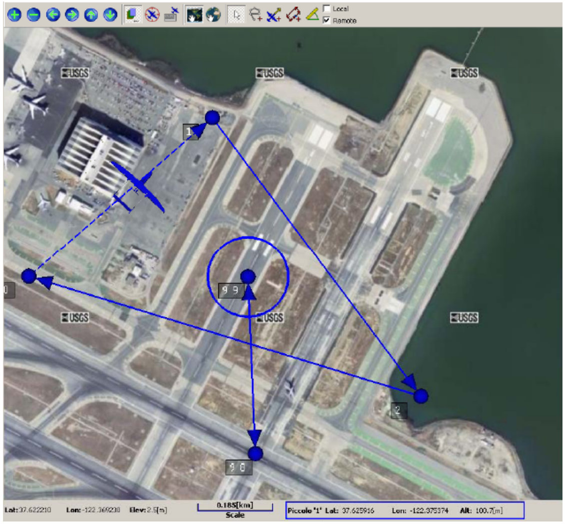
Figure 17. JAUS Software Structure

The JAUS code implements an UDP client and server for testing purposes. The JAUS class was broken down into several sub-classes: conversions, autopilot parser, header, and data. The Conversions class contains the conversions between the JAUS protocol and other data types. The Autopilot Parser class tailors the code to the Piccolo LT autopilot, which reads current autopilot information and commits the information to memory for future use. The Header class handles the data associated with the JAUS header and has three sub-classes: the data control, message properties, and ID classes. The Message Properties field has several portions, which handle bit operations. The ID class contains the node ID, subsystem ID, component ID, and the instance ID, which is needed for both the source and destination of information. The Data class holds the information particular to the data separated from the standard information in the header and composed of the report global position and presence vector sub-classes. The Report Global Position class contains the information from the Report Global Position Message. Information in the Report Global Position is filled by the Autopilot Parser class prior to responding to the JAUS command box. The Presence Vector class handles the operations needed to maintain the presence vector.

4. SAFETY

The overall approach was to have a design that incorporated as many of the safety requirements as possible into a single unit. The Piccolo LT autopilot system satisfied most requirements within the given budget. For those requirements not satisfied by the Piccolo system, an alternative solution was found.

Safety requirement number two (“The system shall provide sufficient information to the judges to ensure that it is operating within the no-fly/altitude boundaries on a continuous basis.”) is met through the main map window of the Piccolo Command Center (Figure 18).



Lat: 37.622210 Lon: -122.369230 Elev: 2.5[m]

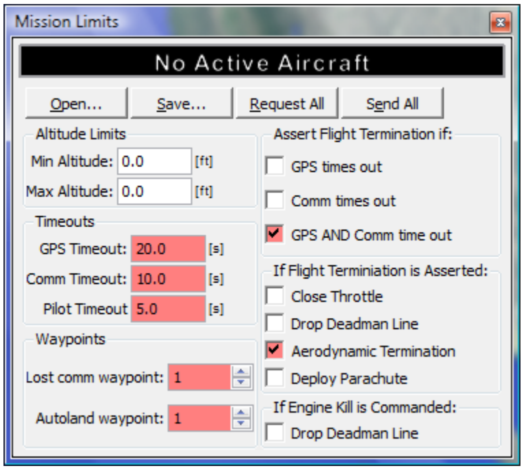
Piccolo '1' Lat: 37.625916 Lon: -122.375374 Alt: 100.7[m]

Figure 18. Main Map Window in Piccolo Command Center

The main map window displays a graphical representation of the aircraft’s two- dimensional position in the main window as well as displaying the latitude and longitude coordinates of the aircraft in the bottom left corner. The aircraft’s elevation is displayed to the right of the latitude and longitude coordinates in the bottom left corner of the main window. In addition, aircraft labels may be used to display the aircraft’s altitude next to the graphical representation of the aircraft in the map window.

Safety requirement number three (“The air vehicle shall be capable of manual override by the safety pilot during any phase of flight.”) is achieved by the Piccolo system, allowing the user to take manual control of the aircraft through the use of a standard RC aircraft transmitter and console cable linking the transmitter commands to the Piccolo Ground Station. However, Team Xipiter chose not to rely solely on the Piccolo system for this requirement. Instead, a servo interface module capable of switching between two sets of telemetry commands was used so that a second radio link could be used to take control of the aircraft. A second radio link of a different frequency (and not a multiple of the primary radio link) was chosen as a backup communication link to the aircraft. This significantly increases the reliability and safety of the system.

Safety requirement number four (“The air vehicle shall automatically return home or terminate flight after loss of transmit signal of more than 30 sec.”) is satisfied through the use of the mission limits console in the Piccolo Command Center (Figure 19).



Altitude Limits
Min Altitude: 0.0 [ft]
Max Altitude: 0.0 [ft]

Timeouts
GPS Timeout: 20.0 [s]
Comm Timeout: 10.0 [s]
Pilot Timeout 5.0 [s]

Waypoints
Lost comm waypoint: 1
Autoland waypoint: 1

Assert Flight Termination if:
□ GPS times out
□ Comm times out
☑ GPS AND Comm time out

If Flight Terminiation is Asserted:
□ Close Throttle
□ Drop Deadman Line
☑ Aerodynamic Termination
□ Deploy Parachute

If Engine Kill is Commanded:
□ Drop Deadman Line

Figure 19. Mission Limits Control Panel

The Piccolo Command Center allows the user to set flight plans based on specific timeouts. By setting a timeout for the communications link and specifying a flight plan, the aircraft will return home in the event of a communications timeout.

Safety requirement number five state: “The air vehicle shall automatically terminate flight after loss of signal of more than 3 minutes.” This requirement is superfluous due to requirement four in that the aircraft has already returned home or terminated flight after only 30 seconds of no communication. Therefore, this requirement has already been met, in essence, because requirement number four has been met.

Safety requirement number six (“The return home system, if installed, shall be capable of activation by the safety pilot.”) is met by the abort command in the Piccolo Command Center. This command is controller specific and can be programmed to perform a variety of tasks.

Safety requirement number seven (“The flight termination system shall be capable of activation by the safety pilot.”) is satisfied in the same manner as the return home system.

Safety requirement number eight specifies flight termination parameters. This flight termination routine is defined in Piccolo’s flight termination as the “aerodynamic termination”. An extra layer of safety was added to the system by installing a backup radio control to the aircraft. This link is on a different spread spectrum frequency and is not a multiple of the main radio frequency.

5. TESTING AND EVALUATION

5.1 Test Bed

While the X-2C airframe was under construction, a kit plane was used to test several of the systems earlier in the year. The Sig Kadet was chosen for simplistic construction, ease in modification, and proper capacity for the systems. These modifications included installing sturdier landing gear, constructing platforms and hatches to mount and access components, and installing a larger engine. The plane was then outfitted with the proper wiring and batteries in order to house the autopilot and camera systems. For these tests, a Black Widow camera was used for its smaller size and weight. Over the course of the year, numerous missions were flown in order to familiarize the team with the Piccolo autopilot as well as to verify XBS and XawkEyes functionality. By utilizing the Sig Kadet test bed, the team was able to perform early systems flight tests, greatly increasing team understanding of the system as a whole and allowing for more time to resolve problems. Also, the team was able to practice flight briefings, prepare test plans, run missions, hold mission de-briefings, and formulate new missions based on previous results.

5.2 Autopilot Testing

In order to insure stable autonomous flight and to become familiar with the autopilot system, autopilot testing had to be performed. This was first done using a Sig Kadet test plane. The plan created for this divided the testing into separate parts that allowed complexity to be added as each task was completed. The first part of the test plan called for a manual flight of the aircraft to trim and test airworthiness. Next, the autopilot was programmed with waypoints in a racetrack pattern in order to test tracking and the gains generated from the simulator. Subsequently, a flight plan was created with varying altitudes to determine how the aircraft tracked altitude changes. The final test involved starting the aircraft in a simple circular pattern and dynamically adding a waypoint while in flight.

5.3 Surveillance System Testing

For testing, the software that accompanied the camera will be used as well as the XCCS. Specifications such as resolution, frame rate, and movement speed will be tested for accuracy and consistency.

After controlling the camera with its accompanied software and XCCS, changing options, and viewing the video, some specifications were validated, but a few problems were also found. As far as resolution and frame rate are concerned, it was found that they were as specified, but there is a significant lag between when an action occurs and when its video appears; the higher the resolution, the higher the lag. Changes in XBS will be required to account for the lag in the imagery.

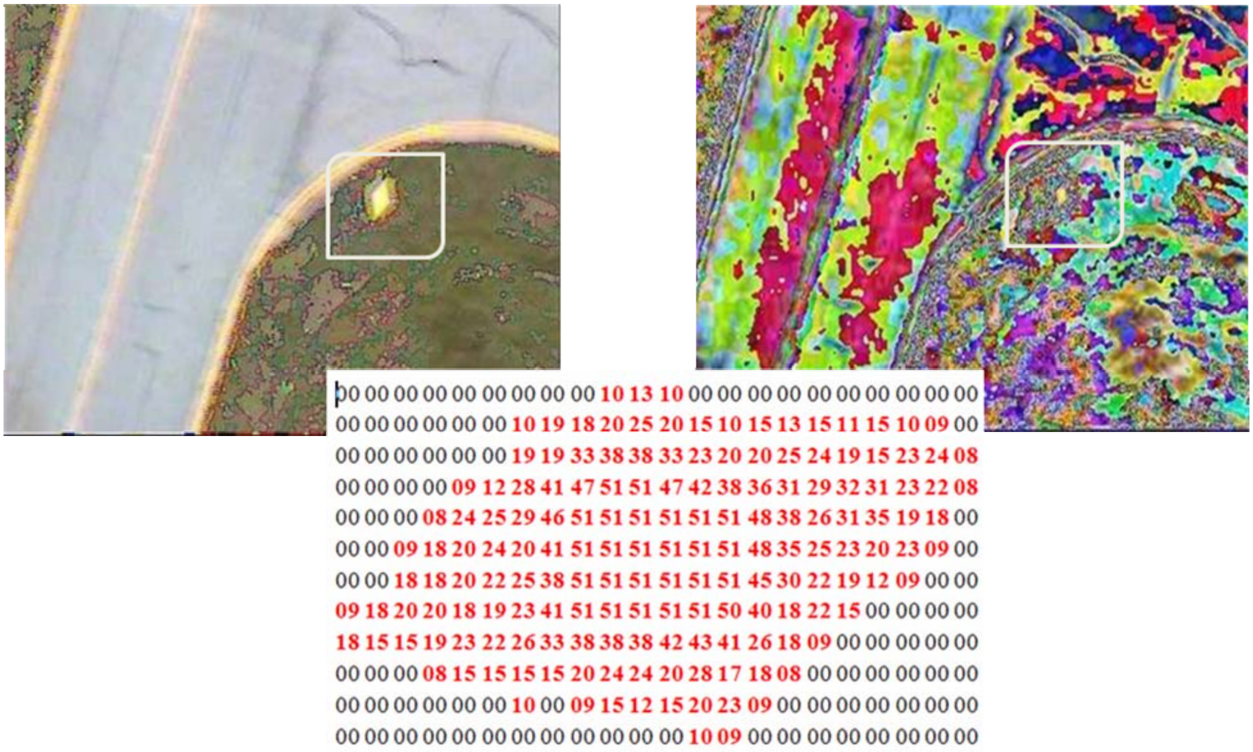
Another problem found was the difficulty in commanding an absolute camera position. The camera has no single command allowing movement to absolute coordinates. The solution is to create a preset with the absolute coordinates, and then command the camera to move to that preset. Common positions will be saved ahead of time, and the camera will be commanded to move between just these presets.

To test the XCCS software, every feature will be tested under different circumstances. Video will be taken with and without communication to the autopilot; joystick control will be tested with and without communication with the camera device, etc.

All modules have been tested and successfully validated. One notable problem was with the lag discovered in the time between sending a movement command and actual camera motion. Further investigation with a network protocol analyzer determined that some of the problem was in the network camera module in XCCS. The problem was that the software took some time to send the command to the camera after the action was received from the joystick. The module was rewritten to remove this portion of the lag, so that as soon as the action is made with the joystick, the camera receives the command. Some lag still exists between the command reception and actual movement due to camera hardware. By minimizing camera movement, the effect of this lag is within tolerance.

5.4 XawkEyes Testing

To test the XawkEyes software, the images gathered from the Black Widow camera during test flights as well as previously gathered images from competition were used. The images were passed through the software with the intermediate steps shown below in Figure 20. The testing showed the XawkEyes Software could successfully identify shapes and colors from images.



```
00 00 00 00 00 00 00 00 00 10 13 10 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 10 19 18 20 25 20 15 10 15 13 15 11 15 10 09 00
00 00 00 00 00 00 19 19 33 38 38 33 23 20 20 25 24 19 15 23 24 08
00 00 00 00 09 12 28 41 47 51 51 47 42 38 36 31 29 32 31 23 22 08
00 00 00 08 24 25 29 46 51 51 51 51 51 51 48 38 26 31 35 19 18 00
00 00 09 18 20 24 20 41 51 51 51 51 51 51 48 35 25 23 20 23 09 00
00 00 18 18 20 22 25 38 51 51 51 51 51 51 45 30 22 19 12 09 00 00
09 18 20 20 18 19 23 41 51 51 51 51 51 50 40 18 22 15 00 00 00 00
18 15 15 19 23 22 26 33 38 38 38 42 43 41 26 18 09 00 00 00 00 00
00 00 00 08 15 15 15 15 20 24 24 20 28 17 18 08 00 00 00 00 00 00
00 00 00 00 00 00 10 00 09 15 12 15 20 23 09 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 10 09 00 00 00 00 00 00 00 00
```

Figure 20. XawkEyes Testing

The first image shows a diamond shape after the edges were sharpened. The second image is the same picture shown in the three-dimensional color space. The diamond shape can still be seen. The bottom image is the color values assigned to the yellow area where the diamond is located.

5.5 XBS Testing

Testing for the XBS software was conducted on the Sig Kadet systems test bed. Targets of known side were placed at known GPS locations. The coordinates were checked via a handheld GPS unit and by holding the autopilot over the location and recording the GPS location. Then the plane flew over the targets and took snapshots with the Black Widow test camera while the appropriate flight and camera data was recorded. By analyzing the pictures and data through XBS and comparing the calculated GPS coordinates and measurements to the known GPS coordinates and measurements (Table 4), the software was evaluated and any error in the XBS software was determined. Latitude and longitude were accurate to within five meters and target size was accurate to within eleven inches.

Table 4. XBS Testing Results

Target 1 - Actual Location: 33.4292 N, 88.8470 W Actual Size: 2.5 x 1.67 ft				
Picture #	**Altitude (ft)**	**XBS Latitude (°)**	**XBS Longitude (°)**	**Size (ft)**
1	236	33.4307	88.8471	3.18 x 2.65
2	268	33.4305	88.8477	2.72 x 2.72
3	264	33.4307	88.8469	2.37 x 2.67
4	200	33.4276	88.8472	1.80 x 2.03
	Average	33.4299	88.8472	2.52 x 2.52
	% Difference	0.002%	0.000%	33.85%
Target 2 - Actual Location: 33.4291 N, 88.8470 W Actual Size: 2.5 x 1.67 ft				
Picture #	**Altitude (ft)**	**XBS Latitude (°)**	**XBS Longitude (°)**	**Size (ft)**
1	236	33.4307	88.8470	2.92 x 2.66
2	268	33.4305	88.8477	2.20 x 1.81
3	264	33.4307	88.8468	2.67 x 1.78
4	200	33.8471	88.8471	2.74 x 2.93
	Average	33.5347	88.8471	2.63 x 2.30
	% Difference	0.315%	0.008%	27.52%

Possible sources of error include the assumption that the coordinate systems originate at the aircraft center of gravity, inaccuracies in the conversion from distances to degrees of latitude and longitude, GPS accuracy, and general numerical inaccuracies propagated through the system.

5.6 JAUS Testing

For testing of the JAUS software, a test case was created for each class evaluating each individual function. After each class passed all of the tests then a test case was written to test the JAUS class overall. Next the JAUS class was added to the UDP client and UDP server to test the validation requirements set by AUVSI. Once the testing was completed with the server, then testing was performed with the provided JAUS Compliance Suite, since this will be the software used at the competition.

The main complication with JAUS was becoming familiar with the JAUS protocol. Reading through the official documentation and other JAUS projects aided in breaking the information into understandable pieces. Another problem encountered was that Java had useful data types helping with the overall implementation, but with no simple way to convert between them. Thus, the Conversions class was created. Also, Java does not support unsigned integers which made it difficult when it came to the RMS values.

5.7 Safety Testing

Testing of our solution to safety requirement number two requires that the GPS and altitude data be correct and visual inspection of the Piccolo Command Center Main Map Window. To verify the GPS and altitude data, the aircraft will be taken to the airfield and the data will be verified against known sources. To verify the GPS data, a Garmin hand-held GPS unit will be used. Altitude information will be verified against geological survey data of the airfield.

Testing of requirement number three will be done during test flights of the aircraft. This requirement will first call for range checks of both radio links before the first take-off of the aircraft. The aircraft will be placed at one location while the transmitter will control the aircraft from 1.5 times the maximum distance the aircraft can be during the competition. The aircraft will then be flown in autopilot mode and during this run, and the autopilot will be disengaged via the Piccolo transmitter. This test setup will be performed again using the backup 2.4GHz transmitter. Demonstration of the remaining requirements can only be performed by simulation due to potential loss of the airframe.

6. ACKNOWLEDGEMENTS

The many strides made by Team Xipiter this year would not have been possible without the aid and guidance of our advisors Mr. Calvin Walker and Dr. Randy Follett. We sincerely thank the Bagley College of Engineering, the Department of Aerospace Engineering, and the Department of Electrical and Computer Engineering at Mississippi State University. We especially appreciate the contributions of Hitech, American Eurocopter, 5-D Systems, Inc.; Hexcel, Mr. and Mrs. Bill Cope, Mr. and Mrs. Doyle Edwards, Schering-Plough, El Paso, Aerodyne, Toshiba, Hill Computing Associates, DR Hobbies, Xoar International, and Aviall.

7. REFERENCES

[1] Canny, J.F.: A variational approach to edge detection. In: AAAI. (1983) 54-58

[2] Laplace, S., Dalmau, M., Roose, P.: Prise en compte de la qualité de service dans la conception et l'exploitation d'applications multimédia réeparties. In: INFORSID. (2006) 815-830

[3] Sobel, I., Feldman, G.: A 3x3 isotropic gradient operator for image processing (1668)

[4] Swain, M.J., Ballard, D.H.: Color indexing. International Journal of Computer Vision 7(1) (1991) 11-32

[5] Gevers, T., Smeulders, A.W.M.: Color based object recognition. In Bimbo, A.D., ed.: ICIAP (1). Volume 1310 of Lecture Notes in Computer Science., Springer (1997) 319-326

[6] Song, J., Cai, M., Lyu, M.R.: Edge color distribution transform: An efficient tool for object detection in images. In: ICPR (1). (2002) 608-612

[7] Schauf, M.L., Aksoy, S., Haralick, R.M.: Model-based shape recognition using recursive mathematical morphology. In: Proceedings of the 14th International Conference on Pattern Recognition. Volume 1., Queensland, Australia, IEEE Computer Society (August 1998) 202

[8] Wolf, Paul R., Elements of Photogrammetry, McGraw-Hill, Inc., New York (1983)